비트패턴을 이용한 고속 워핑 예측

강봉구[†] · 안재형[††]

요 약

본 논문에서는 비트패턴을 이용한 고속 워핑 예측을 제안한다. 워핑 예측은 이웃한 노드들의 움직임 벡터에서 움직임 정보를 가져오는 공간적인 의존성 때문에 최적의 움직임을 찾기 위해서는 반복적인 탐색을 요구한다. 반복적인 탐색에 의해 발생하는 연산량의 증가는 워핑 예측의 사용을 저해하는 큰 이유 중의 하나이다. 본 논문에서 제안한 방법은 비트패턴을 이용해서 움직임 여부를 판단한다. 움직임이 없는 영역에서의 의미없는 움직임 예측을 제거하고, 움직임이 있는 영역에서만 워핑 예측을 수행함으로써 빠른 움직임 예측이 가능하게 된다. 제안된 방법으로 실험한 결과 기존의 워핑 예측을 수행했을 때보다 성능은 근접하면서도 연산량이 75% 이상 감소하였다.

Fast Warping Prediction using Bit-Pattern for Motion Estimation

Bong-Goo Kang[†] and Jae-Hyeong Ahn[††]

ABSTRACT

In this paper, we propose a fast warping prediction using bit-pattern for motion estimation. Because of the spatial dependency between motion vectors of neighboring node points carrying motion information, the optimization of motion search requires an iterative search. The computational load stemming from the iterative search is one of the major obstacles for practical usage of warping prediction. The motion estimation in the proposed algorithm measures whether the motion content of the area is or not, using bit-pattern. Warping prediction using the motion content of the area make the procedure of motion estimation efficient by eliminating an unnecessary searching. Experimental results show that the proposed algorithm can reduce more 75% iterative search while maintaining performance as close as the conventional warping prediction

1. 서 론

통신 기술의 발달로 고속의 네트워크 환경이 구축되었고 이에 멀티미디어 데이터 통신이 가능하게 되었다. 현재의 기술은 멀티미디어 데이터의 실시간 처리를 충분히 만족스럽게 지원하지 못한다. 이것을 해결하고자 멀티미디어 데이터의 표준화 작업과 압축기술에 대한 연구가 활발히 진행되고 있으며, 동영상 압축도 그 중 하나이다. 동영상 압축은 동영상의 높은 시간적 상관관계의 특성을 이용한다. 따라서 프레임간의 시간적 중복성을 제거하는 움직임 예측은 동영상 압축의 가장 핵심적인 부분이다.

현재 움직임 예측에는 간단하고 효과적인 방식인 블록 매칭 알고리즘(block matching algorithm: BMA) [1]이 주로 사용되고 있다. BMA는 H.261/263, MPEG-1/2에 채택되어져 있는 알고리즘이다. BMA는 한 블록 내의 모든 화소들이 동일한 움직임을 갖는다는 가정 하에 한 블록 당 하나의 움직임 벡터(motion vector: MV)를 할당한다. 그러나, 동영상 각각의 프레임을 일정한 규격의 정방형 블록으로 나누어서 움직임 벡터를 찾고 또 움직임을 보상하는 과정에서 블록킹 잡음(blocking artifacts)[2]을 생성하게 되며,

† 준회원, 충북대학교 정보통신학과
†† 정회원, 충북대학교 전기 전자공학부 교수

블록 단위로 움직임을 보상하는 과정에서 블록 경계선에서 불연속적인 움직임이 존재할 때 필연적으로 나타나는 현상이다.

워핑 예측(warping prediction)[2-5]은 제어 격자 보간법(control grid interpolation: CGI)[2]이라고 불리기도 하며 BMA는 CGI의 특별한 예로서 보여질 수 있다. 워핑 예측은 전체 프레임에서 움직임 벡터를 찾는데 있어서 BMA처럼 규격화된 블록으로 나누는 것이 아니라 겹쳐지지 않은 폴리곤(polygon)의 집합으로 그물(mesh) 모양의 형태로 프레임을 분할하게 된다. 워핑 예측은 어떤 규격화된 모양이 아닌 폴리곤의 형태로 프레임을 나누어 움직임 벡터를 찾기 때문에 BMA의 단점인 블록킹 잡음을 생성하지 않는다. 그러나, 워핑 예측은 일정한 폴리곤의 형태를 찾기 위한 무수히 많은 반복적인 연산으로 연산량이 증가하는 단점을 갖고 있다.

본 논문에서는 워핑 예측의 반복적인 연산량을 줄이기 위해서 비트패턴(bit-pattern)을 이용한 고속 워핑 예측 알고리즘을 제안한다. 먼저 비트변환을 통해서 영상을 단순화한 뒤 프레임 내 움직임이 존재하는 영역과 움직임이 없는 영역을 구분한다. 움직임이 없는 영역에서는 반복적인 연산량을 제거하고, 움직임이 탐색된 영역에는 워핑 예측을 수행함으로써 기존의 워핑 예측보다 빠른 움직임 예측을 한다.

2장은 워핑 예측을, 3장은 비트패턴의 기본 개념을 설명하고, 4장은 제안 알고리즘을 설명한다. 5장에서는 실험결과를 통해 성능을 평가하고 6장에서 결론을 맺는다.

2. 워핑 예측

CGI라고도 알려진 워핑 예측은 전체 프레임을 그물이라 불리는 겹쳐지지 않은 폴리곤의 집합으로 나눈다. 그물 구조(mesh structure)를 이루는 워핑 예측에서 움직임 벡터는 노드(node)가 전달하며, 폴리곤 내부의 움직임 필드(motion field)는 공간 변환(spatial transformation)과 둘러싼 노드들의 움직임 벡터로부터 구해진다. 워핑 예측에서 예측되어진 프레임은 식 (1)과 같다. 현재 프레임 이미지 F_n에 대한 예측되어진 프레임 이미지 $\hat{F}_n$은 이전 프레임 이미지 $\widetilde{F}_n$-1에서 찾는다[5].

$$\hat{F}_n(x,y) = \widetilde{F}_{n-1}\big(f(x,y), g(x,y)\big) \qquad (1)$$

$f(x, y)$와 $g(x, y)$가 변형되는 곳은 $\hat{F}_n(x, y)$와 $\widetilde{F}_n(x', y')$ 간의 기하학적 관계에 의해 결정된다. 다음의 두 개의 함수 $f(x, y)$와 $g(x, y)$를 사용하여 각각의 픽셀에 대한 움직임 벡터를 계산할 수 있으며, 양선형 변환(bilinear transform)을 사용할 경우 식 (2)와 같다.

$$\begin{aligned} f(x,y) &= a_1xy + a_2x + a_3y + a_4 \\ g(x,y) &= a_5xy + a_6x + a_7y + a_8 \end{aligned} \qquad (2)$$

변환 계수 $\{a_1, \cdots, a_8\}$은 노드들의 움직임 벡터로부터 계산된다. 폴리곤 형태로 나누는 방법은 양선형 변환이나 아핀 변환(affine transformation), 공간 변환(spatial transform) 등을 사용하며, 폴리곤의 모양은 삼각형이나 사변형 그물 구조 등을 사용할 수 있다. 보통 워핑 예측에서는 폴리곤 모양은 사변형 그물 구조를 선택하고 변환 방법은 양선형 보간법(bilinear interpolation)을 사용하여 움직임을 보상한다[2].

그림 1은 노드에서 움직임 보상의 과정인 폴리곤 매칭을 보여준다. 노드 N_c의 움직임 벡터 MV_c는 움직임 필드에 영향을 준다. 움직임 벡터 MV_c의 변화는 움직임 벡터를 포함하는 움직임 필드를 변화시키고 따라서 움직임 보상을 위한 예측되어진 이미지를 변화시킨다.

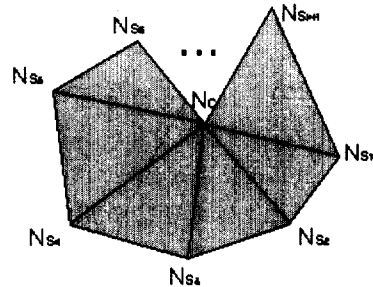



그림 1. 노드에서 움직임 예측을 위한 폴리곤 매칭

워핑 예측은 BMA처럼 규격화된 정방형 블록으로 프레임을 나누는 것이 아니라 움직임의 정도에 따라 폴리곤 형태인 그물 구조로 프레임을 나누기 때문에 BMA 단점인 블록킹 잡음을 생성하지 않는다. 그러나, 이러한 과정에 있어서 무수히 많은 반복적인 연산을 하기 때문에 워핑 예측이 실용적으로 사용되는데 있어서 어려움이 따른다.

3. 비트패턴

비트패턴(bit-pattern)은 해당 블록과 탐색영역의

8bit 화소값을 1bit로 표현한 이진 영상을 의미한다. 이는 탐색과정에서의 계산량을 줄여 빠른 탐색이 이루어지도록 하기 위한 것이다. 탐색블록과 탐색영역의 비트패턴 B는 식 (3)을 통해서 얻어진다[6].

$$B(x,y)=\begin{cases}1, & if\ \ p(x,y)\geq M\\ 0, & otherwise\end{cases} \tag{3}$$

여기서 $p(x, y)$는 탐색블록 또는 탐색영역의 (x, y)번째 화소값을 의미하고, M은 탐색블록의 평균 화소값을 의미한다. 비트패턴은 각각의 블록에 대해서 직관적인 블록의 정보를 나타낸다. 즉, 탐색블록의 비트패턴이 탐색영역 내의 어떤 블록의 비트패턴과 매우 유사하다면 원 영상에서의 두 블록도 유사한 이미지의 특징을 갖고 있다고 볼 수 있다.

(a) F_n-1 프레임 (b) F_n 프레임

그림 2. 비트패턴 이용하여 움직임을 탐색

그림 2에서는 식 (2)을 사용해서 움직임을 탐색한 예를 보여준다. F_n 프레임 블록들의 화소값들을 이진화한 다음 F_n-1 프레임 블록들의 이진 화소값을 비교하여 움직임의 여부를 판단할 수 있다. 즉, 두 프레임간의 이진값 비교를 통해서 움직임의 영역을 빠르게 검색할 수 있다. 비트패턴을 기반으로 움직임을 예측하였을 경우 상당히 많은 계산량을 줄일 수 있다는 장점이 있다. 이는 영상의 표현을 8bit에서 1bit로 이진화 시켰기 때문이다. 원 영상의 화소값을 이진화할 때 M값의 선택에 따라 계산량과 예측 에러가 변화한다.

4. 비트패턴을 이용한 고속 워핑 예측

본 논문에서는 사변형 그물 구조를 갖는 CGI와 양선형 보간법을 기반으로 비트패턴(bit-pattern)을 이용한 고속 움직임 탐색이 가능한 방법을 제안한다. 일반적으로 동영상의 각각의 프레임은 움직임을 가지는 움직임 객체 부분과 움직임이 없는 배경 부분으로 구성되어 있다. 만약, 움직임이 없는 배경부분에 대한 움직임 예측 과정을 간소화할 수 있다면 탐색시간과 계산량을 감소시킬 수 있을 것이다. 따라서 각각의 프레임에서 움직임이 있는 영역과 움직임이 없는 영역을 판단할 필요가 있다. 식 (4)와 같이 움직임 벡터를 (0,0)으로 설정하고 두 블록 사이의 차분인 DBD(displaced block difference)를 계산함으로써 프레임 내의 특징 영역에서 움직임 여부를 판단할 수 있다.

$$DBD(0,0)=\frac{1}{N\times N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\left|F_n(x,y)-F_{n-1}(x,y)\right|, \tag{4}$$

$$\begin{cases}if\ \ DBD(0,0)\geq TH_{motion}, & motion\ searching\\ otherwise, & no\ motion\ searching\end{cases}$$

즉, $DBD(0,0)$이 특정 임계값 보다 작으면 움직임이 없다고 추정할 수 있으므로 그 영역에 대한 움직임 예측 과정을 감소시킬 수 있다[7]. 이를 통해서 움직임이 없는 영역에서의 불필요한 연산을 제거할 수 있다. 일단 움직임이 있는 영역과 움직임이 없는 영역을 구분한 후 다음 단계로 움직임이 없다고 판단되는 영역은 블록으로 폴리곤의 형태가 주어지고 움직임이 있는 영역은 폴리곤 매칭을 수행한다. 즉, DBD 값이 특정 임계값 보다 클 경우 식 (5)을 통해 폴리곤 매칭을 하게 되는 것이다. 왜곡 측정 함수(distortion measure function) D_F는 각각의 노드에서의 후보 움직임 벡터 MV_{ij}의 연산으로 얻어진다.

$$D_{ij}=\sum_{k=1}^{N}\left[\sum_{(x,y)\in P_k}D_f\left\{F_n(x,y)-\widetilde{F}_n\left(f_{MV_{ij}}(x,y),g_{MV_{ij}}(x,y)\right)\right\}\right] \tag{5}$$

본 논문에서 제안한 알고리즘의 과정은 다음과 같고 그림 3은 제안 알고리즘의 처리 과정을 순서도로 나타낸다.

- **1 단계 : 영상의 2진화 단계**

참조 프레임과 이전 프레임을 2진화한다. 영상의 단순화작업을 통해 다음 단계에서 프레임내 움직임 추적을 쉽게 수행하게 해준다.

- **2 단계 : 움직임 추적 단계**

2진화된 프레임에서 움직임을 검색하여 움직임이 있는 영역과 없는 영역을 판단하고 폴리곤 매칭이 필요한 영역과 필요하지 않는 영역으로 나눈다.

• **3 단계 : 폴리곤 매칭이 필요 없는 노드 결정 단계**

움직임이 없다고 판단되는 영역은 폴리곤 매칭을 수행하지 않고 2단계에서 얻어진 블록의 노드를 최종 노드로 결정한다.

• **4 단계 : 폴리곤 매칭을 통한 노드 결정 단계**

움직임이 검출된 영역은 폴리곤 매칭을 통해 최적의 사변형 폴리곤을 찾아 노드의 움직임 벡터를 예측한다.

• **5 단계 : 최종 움직임 벡터 결정 단계**

3단계와 4단계 과정을 거쳐 최종적으로 전체 프레임의 그물 구조를 결정하고 각 노드에서의 움직임 벡터를 예측한다.

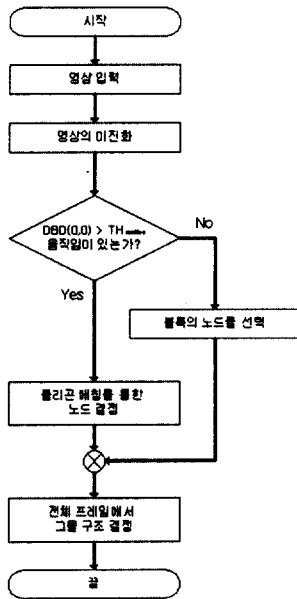
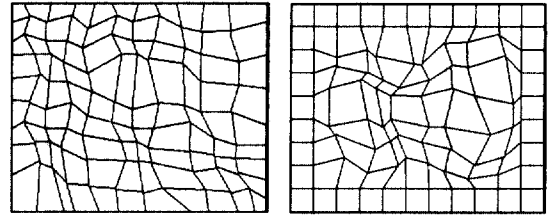



그림 3. 제안 알고리즘의 순서도

워핑 예측과 제안한 알고리즘의 폴리곤 매칭을 한 예를 그림 4에서 보여준다. 움직임이 없는 영역에서의 반복적인 연산의 제거로 전체 연산량을 감소시킬 수 있다.

5. 실험 결과

본 논문에서 제안한 알고리즘을 평가하기 위해서 CIF(common intermediate format: 352288 color pixels) 형식인 Clare영상과 Salesman영상을 각각 1번부터 106번 프레임까지 사용하였다. 성능을 비교

(a) 기존의 워핑 예측

(b) 고속 워핑 예측

그림 4. 전체 프레임에서 알고리즘간의 폴리곤 매칭 비교

하기 위해서 전역 탐색 블록 매칭 알고리즘(full search block matching algorithm: FS-BMA)은 블록 크기를1616으로 하였고, CGI와 제안한 알고리즘은 사변형 그물 구조와 양선형 보간법(bilinear interpolation)을 사용하여 396개의 노드들이 존재하도록 했다. 움직임 예측은 일반적인 동영상 코딩의 참조 프레임의 주기를 적용하여 세 프레임을 간격으로 실시하였다. 본 논문에서는 알고리즘의 비교 분석을 위해 MSE(mean square error)를 구한 다음 식 (6)을 사용해서 PSNR(peak signal to noise ration)을 얻었다.

$$PSNR = 10\log_{10}\frac{255^2}{MSE}[dB] \tag{6}$$

표 1은 CGI와 제안 알고리즘간의 PSNR과 폴리곤 매칭 횟수의 평균값을 비교하였다. 그림 5는 FS-BMA, CGI 와 제안 알고리즘간의 PSNR을, 그림 6은 CGI 와 제안 알고리즘간의 폴리곤 매칭 횟수를 그래프로 비교하였다. 결과값을 통해 본 논문에서 제안한 알고리즘이 CGI에 근접하는 성능을 가지고 있으면서도 연산량은 75%이상 감소시킨 것을 알 수 있다.

표 1. 제안 알고리즘에 대한 성능 평가

	Clare		Salesman	
	CGI	Proposed	CGI	Proposed
Average PSNR	38.63892	38.21402	35.03887	34.61639
Average numbers of polygon matching/ Frame	1062.34	242.86	1090.17	272.20

6. 결 론

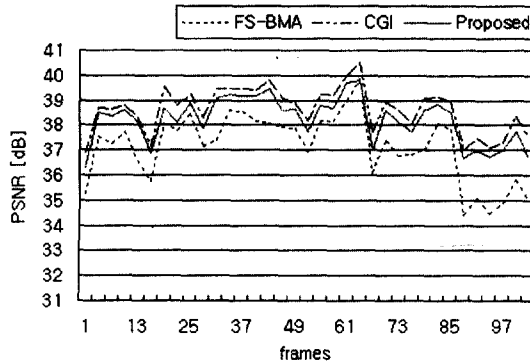
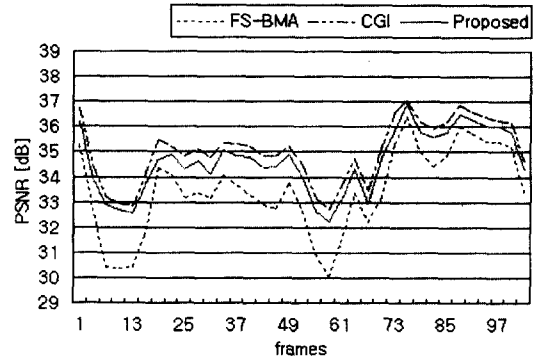
본 논문에서는 비트패턴을 이용한 고속 워핑 예측

(a) Clare (b) Salesman

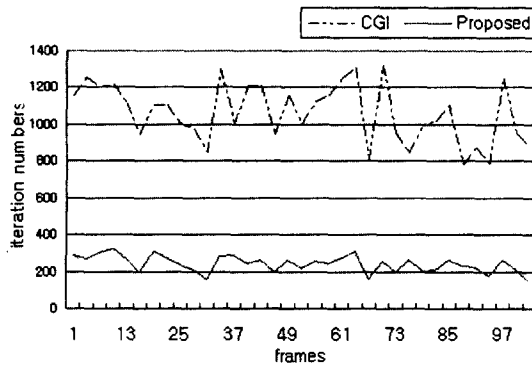
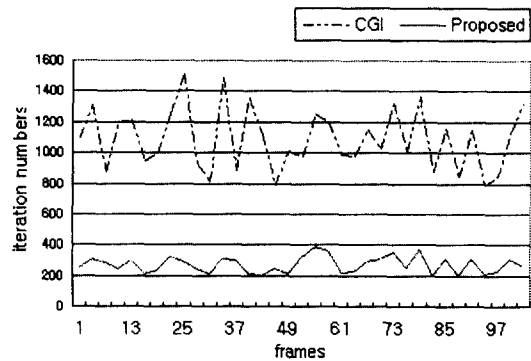
그림 5. PSNR 비교

(a) Clare (b) Salesman

그림 6. 프레임 내에서 폴리곤 매칭의 총 연산 횟수 비교

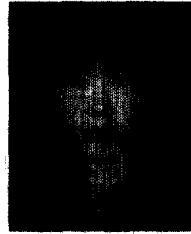
을 제안한다. 기존의 워핑 예측은 프레임 전체에서 폴리곤 매칭을 사용하기 때문에 각 노드의 움직임 벡터를 찾기 위해 무수히 많은 반복적인 연산을 한다. 제안한 알고리즘은 반복적인 연산을 줄이고자 비트변환을 통해 영상을 먼저 이진화 한다. 이진화 하여 단순화된 영상을 가지고 프레임 내의 움직임 정보를 찾아 움직임이 있는 영역과 움직임이 없는 영역을 판단한다. 움직임이 없는 곳에서는 무의미한 폴리곤 매칭을 통한 노드의 움직임 벡터 찾기를 중지하고 블록을 사변형 폴리곤으로 사용한다. 움직임이 있는 영역에 한하여 폴리곤 매칭을 수행하여 노드에서의 움직임 벡터를 추적하여 움직임 예측을 한다. 즉, 움직임이 없는 영역에서는 반복적인 폴리곤 매칭을 제거함으로써 기존의 워핑 예측에 비해 월등한 연산량 감소를 가져왔다.

Clare영상에서 평균 PSNR은 제안한 알고리즘이 FS-BMA보다 1.07450 dB 정도 높고, CGI에 비해서는 0.42429 dB정도 낮았다. Salesman영상에서 평균 PSNR은 제안한 알고리즘이 FS-BMA보다 1.18227 dB정도 높고, CGI에 비해서는 0.44248 dB정도 낮았다. 반복 연산 횟수에서는 Clare영상에서 프레임마다 평균 폴리곤 매칭 횟수는 CGI는 1062.34회이고, 제안한 알고리즘은 242.86회로 77.14%가 감소했으며, Salesman 영상의 경우 CGI는 1090.17회인 반면 제안한 알고리즘은 272.20회로 75.03%의 연산량 감소 효과를 가져왔다. 실험 결과에서 알 수 있듯이 제안 알고리즘은 FS-BMA보다 성능이 우수했고, CGI에 근접한 PSNR을 나타내면서도 폴리곤 매칭 횟수는 평균 75%이상 감소하여 고속 움직임 예측이 가능하였다.

제안 알고리즘은 영상 내의 움직임의 변화가 적은 화상 회의나 화상 전화와 같은 저비트율 전송을 위한 움직임 예측에 효과적으로 적용할 수 있다.

참 고 문 헌

[1] J. R. Jain and A. K. Jain, "Displacement mea-

surement and its application in interframe image coding," *IEEE Trans. Communication*, vol. COM-29, no.12, pp.1799-1808, December 1981
[2] G.J. Sullivan and R.Baker, "Motion compensation for video compression using control grid interpolation," *Proc. IEEE ICASSP-91*, pp.2713-2716, 1991
[3] A. Nosratinia and M.T. Orchard, "Optimal warping prediction for video coding", *1996 IEEE International Conference*, Vol.4, pp.1986-1989, 1996
[4] Dong-Il Chang and Seung-Hwan Kim, "Fast motion search for warping prediction based on node classification," *IEEE*, Vol.2, pp.862-867, 1998
[5] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol.4, no.3, pp.339-356, 1994
[6] J. Feng, K.-T. Mehrpour, and A.E. F. Karbowiak, "Adaptive block matching algorithm for video compression," *IEEE Proc. Vision, Image and Signal Processing*, vol.145 no.3, pp.173-178, 1998
[7] 김진환외, "복잡한 움직임 영역의 예측을 통한 가변 블록 크기 움직임 추정", 신호처리 합동학술대회 논문집, 제12권 제1호, pp.65-68, 1999

강 봉 구

1999년 충북대학교 정보통신공학과(학사)
2001년 충북대학교 정보통신공학과(석사)
2001년~현재 대신증권 전산본부
관심분야 : 영상 통신, 데이터웨어하우징, CRM

안 재 형

1981년 충북대학교 전기공학과(학사)
1983년 한국과학기술원 전기 및 전자 공학과(석사)
1992년 한국과학기술원 전기 및 전자 공학과(박사)
1987~현재 충북대학교 전기 전자 공학부 교수
관심분야 : 영상 통신 및 영상정보처리, 멀티미디어 제작 및 정보제공, 인터넷 통신 및 프로그래밍